80

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/825,753	04/03/2001	Larry Widigen	WDGNP001	7284

42846 7590 01/04/2005

KORBIN VAN DYKE
3343 LITTLE VALLEY ROAD
SUNOL, CA 94586

EXAMINER	
LI, AIMEE J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 01/04/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No.	Applicant(s)
	09/825,753	WIDIGEN, LARRY
	Examiner	Art Unit
	Aimee J Li	2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 17 August 2004 and 06 October 2004.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 21-53 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 21-53 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. New claims 21-53 have been added and considered. Claims 1-20 have been canceled as per Applicant's request.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed on record in the file: Amendment and Extension of time as filed 17 August 2004 and Amendment as filed 06 October 2004

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 21 and 44-49 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chan et al., U.S. Patent No. 6,298,435 in view of Yeager et al., U.S. Patent No. 5,758,112 and further in view of Yung et al., U.S. Patent No. 5,546,554.

5. Referring to claim 21, Chan has taught a method of mapping a plurality of virtual registers to a plurality of physical registers, the method comprising:

a. Providing a plurality of virtual registers, wherein each of the virtual registers comprises physical register address bits (Chan Col.4 lines 20-33); and providing a status indicator for indicating a status of each of the virtual registers (Chan Col.5 lines 5-16);

b. Executing a restore command, the executing the restore command comprising popping the mapping of all of the virtual local registers from the stack to the virtual local registers; and popping the status of all of the virtual local registers from the stack;

c. Binding an argument, the argument binding comprising binding a first virtual register of the virtual registers to a second virtual register of the virtual registers; and binding the status of the first virtual register to the second virtual register;

d. Wherein the argument binding further comprises saving a mapping of the second virtual register onto the stack, saving the status of the second virtual register onto the stack, placing a physical address stored in the first virtual register in the second virtual register, and setting the status of the second virtual register to the status of the first virtual register; and

e. Wherein the argument binding occurs during a call instruction, wherein the call instruction has at least one argument, wherein the first virtual register is used for the at least one argument.

6. Chan has not explicitly taught the method further comprising designating a plurality of virtual registers of the plurality of virtual registers as virtual local registers. Yeager has taught that a subset of the virtual registers which reside in the mapping tables are saved on the branch stack so that a precise restore point for a branch instruction can be restored (Yeager Col. 17 lines 26-38). Here, the logical destination registers of the possible branch instructions are saved, which can be considered "local" to each branch instruction. Because it is necessary to restore a precise state upon return from a branch instruction rather than an imprecise state so that

processing results are correct, one of ordinary skill in the art would have found it obvious to modify the processor of Chan to designate a plurality of the virtual registers as "local" so that the registers can be used to save and restore the state of branch instructions on a branch stack.

7. Chan in view of Yeager has executing a save command, comprising saving the mapping of all virtual local registers onto a stack (see Yeager Col.17 lines 26-38). Chan in view of Yeager has not explicitly taught the saving a status as indicated by the status indicator for each of the virtual local registers onto the stack. However, Chan has taught the use of status indicators to show if a virtual register is valid and has been written into or not (see Chan Col.5 lines 5-16). One of ordinary skill in the art would have recognized that in order to restore a "precise state" as necessitated by Yeager (see Yeager Col.17 lines 26-38), one would have to know which parts of the state were valid and which were not. Furthermore, because the saving of the local virtual registers onto the stack records the changes made to the local virtual registers (see Yeager Col.17 lines 35-38), one would have recognized that the status indicators of Chan would have been updated to reflect the new assignments and writes, as that is the purpose of the indicators (see Chan Col.5 lines 5-16). Therefore, one of ordinary skill in the art would have found it obvious to modify the processor of Chan in view of Yeager to include saving the virtual register status indicators on the stack when saving the local virtual registers so that a correctly precise state can be restored from the stored information. Chan in view of Yeager has not explicitly taught wherein the save command further comprises setting the status of all virtual local registers to "clean". However, one of ordinary skill in the art would have recognized that in a context switch, such as in the execution of a branch instruction (see Yeager Col.17 lines 26-38), there is no purpose to preserving the state of the registers if they are not going to be overwritten in the

new state and thus necessitating a restore of the previous state. Therefore, because the status indicators of the virtual registers dictate whether data new data can be written to them without overwriting valid data (see Chan Col.5 lines 5-16), one of ordinary skill in the art would have found it obvious to set the status indicators to "clean" after executing a context switch so that the new registers are able to be written to and thus justifying the preservation of their previous values.

8. Chan in view of Yeager executing a restore command, comprising popping the mapping of all virtual local registers from the stack to the virtual local registers (see Col.17 lines 5-13, 26-38). Chan in view of Yeager has not explicitly taught wherein executing a restore command further comprises popping the status of all virtual local registers from the stack. However, Official Notice is taken that a stack generally functions as temporary storage, with data being pushed on to it to temporarily save it, and later popped off to restore a previous state. Therefore, because Chan in view of Yeager has taught the pushing of the mapping of local virtual registers along with their corresponding statuses (see Yeager Col.17 lines 26-38), it would have been obvious to one of ordinary skill in the art to pop the statuses corresponding to the local virtual registers so that a previous state could be fully restored.

9. Chan in view of Yeager has not explicitly taught wherein the method further comprises:

a. Binding a first virtual register of the plurality of virtual registers to a second virtual register of the plurality of virtual registers.

b. Binding the status of the first virtual register to a second virtual register.

10. However, Yung has taught when a virtual register is a destination register of an instruction, a new mapping for the virtual register is created so that the original physical register

corresponding to the original virtual register is not overwritten, effectively creating a new virtual register that is identical to the original virtual register except that it is mapped to a new physical register (see Col.8 lines 10-40). Because having data overwritten can cause incorrect processing results which are unacceptable in a processor, one of ordinary skill in the art would have found it obvious to modify the process taught by Chan in view of Yeager to create a new mapping for a virtual register when it is the destination of an instruction, so that data in the physical register of the original mapping is not overwritten, thus avoiding potential incorrect results.

11. Furthermore, Chan in view of Yeager has taught the association of status indicators with virtual registers, and their subsequent storage when the virtual registers are saved during context switches because everything associated with that register needs to be saved to restore the precise state (Yeager Col.17 lines 26-38). One of ordinary skill in the art would have recognized that because Yung is creating a new mapping to be an identical virtual register to an original so it appears the same to the instruction, everything associated with the register needs to be mirrored in the new virtual register so that it is the same in everyway. Therefore, one of ordinary skill in the art would have found it obvious to modify the processor of Chan in view of Yeager with the teachings of Yung so that when binding a virtual register to a new virtual register the associated status is bound also so that a precise copy is created, allowing a precise state to be restored if needed.

12. Chan in view of Yeager in further view of Yung wherein the binding comprises placing a physical address stored in the first virtual register in the second virtual register and setting the status of the second virtual register to the status of the first virtual register (see Yung Col.8 lines 10-40).

13. Chan in view of Yeager in further view of Yung has taught the method as recited in claim 11, wherein the binding further comprises:

a. Saving the mapping of the first virtual register onto the stack cache (see Yeager Col.17 lines 26-38),

b. Saving the status of the first virtual local register onto the stack cache (see Yeager Col.17 lines 26-38).

14. Chan in view of Yeager in further view of Yung wherein the binding occurs during a call instruction, wherein the call instruction has at least one argument, wherein the first virtual register is used for the at least one argument. Here, a branch instruction can be considered a "call" instruction in that it executes code at a new location, and requires that an operand be supplied, generally via a register, to provide an offset. Because Chan in view of Yeager in view of Yung has taught that the binding operation involves saving virtual registers to the stack in response to a branch instruction (see Yeager Col.6 lines 35-60), one of ordinary skill in the art would have found it obvious to consider a branch instruction as a call instruction.

15. Referring to claim 44, Chan has taught a method comprising

a. Decoding an instruction (Chan 109 of Fig.1);

b. Maintaining a mapping of virtual registers to physical registers (Chan Col.4 lines 20-33);

c. If the instruction is a save instruction, then executing a save command (Chan Col.17 lines 26-38);

d. If the instruction is a restore instruction, then executing a restore command (Chan Col.17 lines 5-13, 26-38); and

e. Wherein

i. The executing of the save command comprises saving the mapping of all of the virtual registers onto a stack (Chan Col.17 lines 26-38), and

ii. The executing of the restore command comprises popping the mapping of all of the virtual registers from the stack to the virtual local registers (Chan Col.17 lines 5-13, 26-38).

16. Chan has taught a subset of the virtual registers being virtual local registers. Yeager has taught that a subset of the virtual registers being virtual local registers (Yeager Col.17 lines 26-38). Here, the logical destination registers of the possible branch instructions are saved, which can be considered "local" to each branch instruction. Because it is necessary to restore a precise state upon return from a branch instruction rather than an imprecise state so that processing results are correct, one of ordinary skill in the art would have found it obvious to modify the processor of Chan to designate a plurality of the virtual registers as "local" so that the registers can be used to save and restore the state of branch instructions on a branch stack.

17. Referring to claim 45, Chan in view of Yeager has not explicitly taught saving status indicators corresponding to all of the virtual local registers onto the stack. However, Chan has taught the use of status indicators to show if a virtual register is valid and has been written into or not (see Chan Col.5 lines 5-16). One of ordinary skill in the art would have recognized that in order to restore a "precise state" as necessitated by Yeager (see Yeager Col.17 lines 26-38), one would have to know which parts of the state were valid and which were not. Furthermore, because the saving of the local virtual registers onto the stack records the changes made to the

local virtual registers (see Yeager Col.17 lines 35-38), one would have recognized that the status indicators of Chan would have been updated to reflect the new assignments and writes, as that is the purpose of the indicators (see Chan Col.5 lines 5-16). Therefore, one of ordinary skill in the art would have found it obvious to modify the processor of Chan in view of Yeager to include saving the virtual register status indicators on the stack when saving the local virtual registers so that a correctly precise state can be restored from the stored information.

18. In addition, Chan in view of Yeager has not explicitly taught wherein executing a restore command further comprises popping the status indicators corresponding to all of the virtual local registers from the stack. Official Notice is taken that a stack generally functions as temporary storage, with data being pushed on to it to temporarily save it, and later popped off to restore a previous state. Therefore, because Chan in view of Yeager has taught the pushing of the mapping of local virtual registers along with their corresponding statuses (see Yeager Col.17 lines 26-38), it would have been obvious to one of ordinary skill in the art to pop the statuses corresponding to the local virtual registers so that a previous state could be fully restored.

19. Referring to claim 46, Chan in view of Yeager has taught the method as recited in claim 45, but has not explicitly taught wherein the executing of the save command further comprises setting the status indicators corresponding to all of the virtual local registers to clean after the saving of the status indicators corresponding to all of the virtual local registers onto the stack. However, one of ordinary skill in the art would have recognized that in a context switch, such as in the execution of a branch instruction (see Yeager Col.17 lines 26-38), there is no purpose to preserving the state of the registers if they are not going to be overwritten in the new state and thus necessitating a restore of the previous state. Therefore, because the status indicators of the

virtual registers dictate whether data new data can be written to them without overwriting valid data (see Chan Col.5 lines 5-16), one of ordinary skill in the art would have found it obvious to set the status indicators to "clean" after executing a context switch so that the new registers are able to be written to and thus justifying the preservation of their previous values.

20. Referring to claim 47, Chan has taught a processing device including:

a. An instruction decoder adapted to decode an instruction (Chan 109 of Fig.1);

b. A plurality of physical registers (see Col.4 lines 20-33);

c. A plurality of virtual registers, each of the virtual registers comprising physical register address bits (see Col.4 lines 20-33) and;

d. Wherein

i. If the instruction decoder decodes a save instruction, then executing a save command, the executing of the save command comprising saving a mapping of all of the virtual registers onto a stack (Chan Col.17 lines 26-38), and

ii. If the instruction decoder decodes a restore instruction, then executing a restore command, the executing of the restore command comprising popping the mapping of all of the virtual registers from the stack to the virtual registers (Chan Col.17 lines 5-13, 26-38).

21. Chan has taught a subset of the virtual registers being virtual local registers. Yeager has taught that a subset of the virtual registers being virtual local registers (Yeager Col.17 lines 26-38). Here, the logical destination registers of the possible branch instructions are saved, which

can be considered "local" to each branch instruction. Because it is necessary to restore a precise state upon return from a branch instruction rather than an imprecise state so that processing results are correct, one of ordinary skill in the art would have found it obvious to modify the processor of Chan to designate a plurality of the virtual registers as "local" so that the registers can be used to save and restore the state of branch instructions on a branch stack.

22. Referring to claim 48, Chan has taught

a. A plurality of status indicators, each of the status indicators corresponding to a respective one of the virtual registers (see Chan Col.5 lines 5-16); and

b. Wherein

i. The executing of the save command further comprises saving the status indicators corresponding to all of the virtual registers onto the stack (Chan Col.17 lines 26-38), and

ii. The executing of the restore command further comprises popping the status indicators corresponding to all of the virtual registers from the stack (Chan Col.17 lines 5-13, 26-38).

23. Chan has taught virtual local registers. Yeager has taught virtual local registers (Yeager Col.17 lines 26-38). Here, the logical destination registers of the possible branch instructions are saved, which can be considered "local" to each branch instruction. Because it is necessary to restore a precise state upon return from a branch instruction rather than an imprecise state so that processing results are correct, one of ordinary skill in the art would have found it obvious to modify the processor of Chan to designate a plurality of the virtual registers as "local" so that the registers can be used to save and restore the state of branch instructions on a branch stack.

24. Referring to claim 49, Chan in view of Yeager has taught the method as recited in claim 49, but has not explicitly taught wherein the executing of the save command further comprises setting the status indicators corresponding to all of the virtual local registers to clean after the saving of the status indicators corresponding to all of the virtual local registers onto the stack. However, one of ordinary skill in the art would have recognized that in a context switch, such as in the execution of a branch instruction (see Yeager Col.17 lines 26-38), there is no purpose to preserving the state of the registers if they are not going to be overwritten in the new state and thus necessitating a restore of the previous state. Therefore, because the status indicators of the virtual registers dictate whether data new data can be written to them without overwriting valid data (see Chan Col.5 lines 5-16), one of ordinary skill in the art would have found it obvious to set the status indicators to "clean" after executing a context switch so that the new registers are able to be written to and thus justifying the preservation of their previous values.

25. Claims 22-31 and 34-43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chan et al., U.S. Patent No. 6,298,435 in view of Yeager et al., U.S. Patent No. 5,758,112 and further in view of Yung et al., U.S. Patent No. 5,546,554 and in further view of Microsoft's Computer Dictionary The Comprehensive Standard for Business, School, Library, and Home Second Edition ©1994.

26. Referring to claims 22 and 34, with claim 22 as exemplary, a method comprising:

a. Decoding an instruction (109 of Fig.1);

b. Wherein the argument binding comprises

i. Each of the virtual registers comprising physical register address bits (see Chan Col.4 lines 20-33), and

ii. Wherein each of the status indicators corresponds to a respective one of the virtual registers (see Chan Col.5 lines 5-16); and

c. Wherein the first virtual register is used for the argument and the second virtual register is used as a formal parameter.

27. Chan has not explicitly taught wherein the method further comprises:

a. Copying a first virtual register of a plurality of virtual registers to a second virtual register of the virtual registers,

b. Copying a first status indicator of a plurality of status indicators to a second status indicator of the status indicators,

28. However, Yung has taught when a virtual register is a destination register of an instruction, a new mapping for the virtual register is created so that the original physical register corresponding to the original virtual register is not overwritten, effectively creating a new virtual register that is identical to the original virtual register except that it is mapped to a new physical register (see Col.8 lines 10-40). Because having data overwritten can cause incorrect processing results which are unacceptable in a processor, one of ordinary skill in the art would have found it obvious to modify the process taught by Chan in view of Yeager to create a new mapping for a virtual register when it is the destination of an instruction, so that data in the physical register of the original mapping is not overwritten, thus avoiding potential incorrect results.

29. Furthermore, Chan in view of Yeager has taught the association of status indicators with virtual registers, and their subsequent storage when the virtual registers are saved during context switches because everything associated with that register needs to be saved to restore the precise state (Yeager Col.17 lines 26-38). One of ordinary skill in the art would have recognized that

because Yung is creating a new mapping to be an identical virtual register to an original so it appears the same to the instruction, everything associated with the register needs to be mirrored in the new virtual register so that it is the same in everyway. Therefore, one of ordinary skill in the art would have found it obvious to modify the processor of Chan in view of Yeager with the teachings of Yung so that when binding a virtual register to a new virtual register the associated status is bound also so that a precise copy is created, allowing a precise state to be restored if needed.

30. Chan in view of Yeager in further view of Yung has taught if the instruction is a call instruction, then binding an argument of the call instruction and wherein the first virtual register is used for the argument. Here, a branch instruction can be considered a "call" instruction in that it executes code at a new location, and requires that an operand be supplied, generally via a register, to provide an offset. Because Chan in view of Yeager in view of Yung has taught that the binding operation involves saving virtual registers to the stack in response to a branch instruction (Yeager Col.6 lines 35-60), one of ordinary skill in the art would have found it obvious to consider a branch instruction as a call instruction.

31. Chan in view of Yeager in further view of Yung has not explicitly taught the second virtual register is used as a formal parameter. However, Microsoft has taught on pages 61-62 that a call instruction often passes multiple parameters. A person of ordinary skill in the art at the time the invention was made would have recognized that passing multiple parameters allows these values to be used and modified (Microsoft page 62, thereby ensuring the data is correct and available for use later in the program. Therefore, it would have been obvious to a person of

ordinary skill in the art at the time the invention was made to incorporate the multiple parameters of Microsoft in the device of Chan, Yeager, and Yung to ensure correct data.

32. Referring to claims 23 and 35, with claim 23 as exemplary, Chan has not explicitly taught where the method further comprises:

a. Mapping a virtual register from an old physical register to a new physical register, when the virtual register is a destination virtual register of an instruction being decoded,

b. Placing an address of the old physical register in an instruction retirement list related to the instruction being decoded if an only if the status indicator indicates that the virtual register is not clean.

33. However, Yeager has taught the updated mapping of logical registers to new physical registers after every instruction is decoded (see Col.6 lines 41-45), and the saving of the old physical register, if it has been written to (i.e. not clean) (see Col.7 line 64 – Col.8 line 6), to an active list so that it can be restored later if needed (see Col.6 lines 45-50 and Col.7 lines 54-63). One of ordinary skill in the art would have recognized that it is desirable to increase processing speed in a microprocessor, while keeping complexity to a minimum. Renaming in this manner simplifies dependency checking by making it unambiguous and increases processing speed by reducing the penalty associated with branch mispredictions or exceptions (see Col.6 lines 45-60). Therefore, one of ordinary skill in the art would have found it obvious to modify the processor of Chan to place the address an old physical register on an active list when its associated logical register is mapped to a new register and is not clean as taught by Yeager, so that the speed of the microprocessor is increased and the complexity of dependency checking decreased.

34. Referring to claims 24, 27, 36, and 39, with claim 24 as exemplary, Chan in view of Yeager has taught saving a physical register address held in a virtual register to a stack (see Yeager Col.17 lines 26-38). Chan in view of Yeager has not explicitly taught saving a status of the virtual register indicated by the status indicator to a stack and setting to clean at least status of the virtual register. However, Chan has taught the use of status indicators to show if a virtual register is valid and has been written into or not (see Chan Col.5 lines 5-16). One of ordinary skill in the art would have recognized that in order to restore a "precise state" as necessitated by Yeager (see Yeager Col.17 lines 26-38), one would have to know which parts of the state were valid and which were not. Furthermore, because the saving of the local virtual registers onto the stack records the changes made to the local virtual registers (see Yeager Col.17 lines 35-38), one would have recognized that the status indicators of Chan would have been updated to reflect the new assignments and writes, as that is the purpose of the indicators (see Chan Col.5 lines 5-16). Therefore, one of ordinary skill in the art would have found it obvious to modify the processor of Chan in view of Yeager to include saving the virtual register status indicators on the stack when saving the local virtual registers so that a correctly precise state can be restored from the stored information.

35. Furthermore, one of ordinary skill in the art would have recognized that in a context switch, such as in the execution of a branch instruction (see Yeager Col.17 lines 26-38), there is no purpose to preserving the state of the registers if they are not going to be overwritten in the new state and thus necessitating a restore of the previous state. Therefore, because the status indicators of the virtual registers dictate whether data new data can be written to them without overwriting valid data (see Chan Col.5 lines 5-16), one of ordinary skill in the art would have

found it obvious to set the status indicators to "clean" after executing a context switch so that the new registers are able to be written to and thus justifying the preservation of their previous values.

36. Referring to claims 25, 28, 37, and 40, with claim 37 as exemplary, Chan in view of Yeager has not explicitly taught wherein the method further comprises setting the status of a virtual register to not clean when the virtual register is mapped to a new physical register. However, Chan in view of Yeager has taught the mapping of a virtual register to a new physical register when it has a status of not clean (see Yeager Col.17 lines 26-38). One of ordinary skill in the art would have recognized that this updating of the mapping does not affect the status of the virtual register as other instructions may still be pointing to it, but rather only affects where the data that is mapped with that virtual register is stored. Furthermore, changing the status of the virtual register could prevent the register from being saved or restored during a context switch, producing incorrect results. Therefore one of ordinary skill in the art would have found it obvious to set the status of the virtual register having an updated mapping to "not clean" so that the status of the virtual register remains unchanged, and undesirable results are avoided.

37. Referring to claims 26 and 38, with claim 26 as exemplary, Chan has not explicitly taught a subset of the plurality of virtual registers are virtual local registers. However, Yeager has taught that a subset of the virtual registers which reside in the mapping tables are saved on the branch stack so that a precise restore point for a branch instruction can be restored (see Col.17 lines 26-38). Here, the logical destination registers of the possible branch instructions are saved, which can be considered "local" to each branch instruction. Because it is necessary to restore a precise state upon return from a branch instruction rather than an imprecise state so that processing results are correct, one of ordinary skill in the art would have found it obvious to modify the processor of Chan to designate a plurality of the virtual registers as "local" so that the registers can be used to save and restore the state of branch instructions on a branch stack.

38. Referring to claims 29 and 41, with claim 41 as exemplary, Chan in view of Yeager has taught executing a restore command, comprising popping the mapping of all virtual local registers from the stack to the virtual local registers (see Col.17 lines 5-13, 26-38). Chan in view of Yeager has not explicitly taught wherein executing a restore command further comprises popping the status of all virtual local registers from the stack. However, Official Notice is taken that a stack generally functions as temporary storage, with data being pushed on to it to temporarily save it, and later popped off to restore a previous state. Therefore, because Chan in view of Yeager has taught the pushing of the mapping of local virtual registers along with their corresponding statuses (see Yeager Col.17 lines 26-38), it would have been obvious to one of ordinary skill in the art to pop the statuses corresponding to the local virtual registers so that a previous state could be fully restored.

39. Referring to claims 30 and 42, with claim 30 as exemplary, Chan in view of Yeager has taught selectively executing the restore command if the instruction is a return instruction. Here, a branch instruction can be considered a "return" instruction in that it executes code at the location after the calling instruction. Because Chan in view of Yeager in view of Yung has taught that the restore command involves popping the mapping of all the virtual local registers from the stack (see Yeager Col.6 lines 35-60), one of ordinary skill in the art would have found it obvious to consider a branch instruction as a return instruction. For more information, please see the definition of return (Microsoft page 340).

40. Referring to claims 31 and 43, with claim 31 as exemplary, Chan in view of Yeager in further view of Yung has taught wherein subsequent the binding when the first virtual register is a destination register, the first virtual register is assigned a physical register address which is different than a physical register address stored in the second virtual register (see Yung Col.8 lines 10-40). Here, it is inherent that the new physical address is different than the original physical address as otherwise the data in the original would be overwritten, which is what Yung is avoiding.

41. Claims 32-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chan et al., U.S. Patent No. 6,298,435 in view of Yeager et al., U.S. Patent No. 5,758,112 and further in view of Yung et al., U.S. Patent No. 5,546,554 and in further view of Microsoft's Computer Dictionary The Comprehensive Standard for Business, School, Library, and Home Second Edition ©1994 (herein referred to as Microsoft) and in further view of Shintani et al., U.S. Patent Number 4,736,288 (herein referred to as Shintani). Chan in view of Yung and in further view of Yeager and in further view of Microsoft has not taught

a. Wherein before the assignment of the first physical register address to the first virtual register, a corresponding first physical register status is "free" (Applicant's claim 32).

b. Wherein after the assignment of the first physical register address to the first virtual register, the corresponding first physical register status is "waiting" (Applicant's claim 33).

42. Shintani has taught

a. Wherein before the assignment of the first physical register address to the first virtual register, a corresponding first physical register status is "free" (Applicant's claim 32) (Shintani column 6, lines 19-30 and 37-41; column 7, lines 25-29; and column 8, lines 1-7).

b. Wherein after the assignment of the first physical register address to the first virtual register, the corresponding first physical register status is "waiting" (Applicant's claim 33) (Shintani column 6, lines 19-30 and 37-41; column 7, lines 25-29; and column 8, lines 1-7).

43. A person of ordinary skill in the art at the time the invention was made would have recognized that the physical register statuses of Shintani ensure that data is not overwritten or read in a physical register before the data indicated by the virtual mapping has completely finished being written, thereby ensuring data coherency. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the physical register statuses of Shintani in the device of Chan in view of Yung and in further view of Yeager and in further view of Microsoft to ensure data coherency.

44. Claims 50-53 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chan et al., U.S. Patent No. 6,298,435 in view of Yeager et al., U.S. Patent No. 5,758,112 and further in view of Yung et al., U.S. Patent No. 5,546,554 and in further view of Shintani et al., U.S. Patent Number 4,736,288 (herein referred to as Shintani). Chan in view of Yung and in further view of Yeager has not taught

a. A plurality of physical register status indicators, each of the physical register status indicators corresponding to a respective one of the physical registers (Applicant's claim 50); and

b. Wherein each of the physical register status indicators represents a selected one of a plurality of physical register states, and the physical register states include "free", "waiting", and "valid" (Applicant's claim 50).

c. Wherein physical registers available for mapping to virtual registers are represented as "free" in the corresponding physical register status indicators (Applicant's claim 51).

d. Wherein physical register status indicators transition to representing "waiting" when the corresponding physical registers are mapped to virtual registers (Applicant's claim 52).

e. Wherein physical register status indicators transition to representing "valid" when the corresponding physical registers are written (Applicant's claim 53).

45. Shintani has taught

a. A plurality of physical register status indicators, each of the physical register status indicators corresponding to a respective one of the physical registers (Applicant's claim 50) (Applicant's claim 32) (Shintani column 6, lines 19-30 and 37-41; column 7, lines 25-29; and column 8, lines 1-7); and

b. Wherein each of the physical register status indicators represents a selected one of a plurality of physical register states, and the physical register states include "free", "waiting", and "valid" (Applicant's claim 50) (Applicant's claim 32) (Shintani column 6, lines 19-30 and 37-41; column 7, lines 25-29; and column 8, lines 1-7).

c. Wherein physical registers available for mapping to virtual registers are represented as "free" in the corresponding physical register status indicators (Applicant's claim 51) (Applicant's claim 32) (Shintani column 6, lines 19-30 and 37-41; column 7, lines 25-29; and column 8, lines 1-7).

d. Wherein physical register status indicators transition to representing "waiting" when the corresponding physical registers are mapped to virtual registers (Applicant's claim 52) (Applicant's claim 32) (Shintani column 6, lines 19-30 and 37-41; column 7, lines 25-29; and column 8, lines 1-7).

e. Wherein physical register status indicators transition to representing "valid" when the corresponding physical registers are written (Applicant's claim 53) (Applicant's claim 32) (Shintani column 6, lines 19-30 and 37-41; column 7, lines 25-29; and column 8, lines 1-7).

46. A person of ordinary skill in the art at the time the invention was made would have recognized that the physical register statuses of Shintani ensure that data is not overwritten or read in a physical register before the data indicated by the virtual mapping has completely finished being written, thereby ensuring data coherency. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the physical register statuses of Shintani in the device of Chan in view of Yung and in further view of Yeager and in further view of Microsoft to ensure data coherency.

Response to Arguments

47. Examiner withdraws claim objections in favor of the new claims.

48. Examiner withdraws 35 USC § 112 in favor of the new claims.

49. Applicant's arguments with respect to claims 22-43 and 50-53 have been considered but are moot in view of the new ground(s) of rejection.

50. Applicant's arguments filed 17 August 2004 in regards to claim 21 and 44-49 have been fully considered but they are not persuasive.

51. Applicant argues on pages 24-25 in essence "...the saved destination register mappings have no correspondence to the claimed virtual local registers." This has not been found persuasive. The distinction cited by Applicant is not in the claims. While, the claims are read in view of the specification, no limitations can be read into the claims from the specification. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., the specifics of the virtual local register) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

52. Applicant argues in essence on pages 25-27 regarding the specifics of "binding" in the claims. This has not been found persuasive. The distinction between Applicant's binding and the cited references are not explicitly in the claims. While, the claims are read in view of the specification, no limitations can be read into the claims from the specification. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., the specifics to binding) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

53. Applicant argues in essence on page 28 "...the claimed call instruction argument is different...Thus applicants claimed 'call instructions' 'argument' has not correspondence to a branch instruction offset operand." This has not been found persuasive. This difference is not claimed. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., the call instruction specifics) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Conclusion

54. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

55. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

56. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aimee J Li whose telephone number is (571) 272-4169. The examiner can normally be reached on M-T 7:30am-5:00pm.

57. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

58. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

RICHARD L. ELLIS
PRIMARY EXAMINER

AJL
Aimee J. Li
22 December 2004